Documentation for dynamic_img_anim_maxwell.py

This Python code simulates and visualizes electromagnetic wave propagation using the finite-difference time-domain (FDTD) method. The code leverages the `numpy` library for numerical operations and `matplotlib` for creating animations of the electric field over time.

Overview of Key Variables and Constants

- **num_timesteps**: The total number of time steps for which the simulation will run. In this case, it is set to 100.
- **n**: The spatial resolution of the simulation grid, set to 1000.
- **epsilon**: The permittivity of free space, used in electromagnetic calculations.
- **mew**: The permeability of free space, another fundamental constant in electromagnetics.
- **Ez, Hy, Hx**: These are arrays representing the electric field (Ez) and magnetic fields (Hy, Hx) in the simulation grid.
- **c**: The speed of light in the medium, calculated from the permittivity and permeability.
- **S**: A stability factor used in the time step calculation.
- **dx**: The spatial step size for the grid.
- **dt**: The time step size, calculated based on stability criteria.
- **lambd**: The wavelength, which influences the frequency of the wave.
- **omega**: The angular frequency of the wave.
- **Ca, Da, Cb, Db**: Coefficients used in updating the field values during the simulation.

Code Functionality

The code begins by initializing the simulation parameters and setting up the grid arrays for electric and magnetic fields. The main part of the code is a loop that iterates through the specified number of time steps, updating the field values at each step based on the FDTD method.

Step-by-Step Explanation

1. Initialization:
    - The electric field `Ez` is initialized, and an impulse is set at the center of the grid at the first time step.
2. Field Update:
    - The magnetic fields `Hy` and `Hx` are updated based on the current electric field values.
    - The electric field `Ez` is then updated using the values of `Hy` and `Hx`.
3. Animation Frame Creation:
    - Every 5 time steps, the current state of the electric field is captured as a frame for the animation.

Example of Usage

To run this script, simply execute it in an environment where `numpy` and `matplotlib` are installed. The resulting animation will display the propagation of the electromagnetic wave over the grid, providing a visual representation of the wave dynamics.

Conclusion

This code serves as a useful tool for visualizing electromagnetic wave propagation and can be modified to explore different configurations and parameters in the simulation. Happy coding!